PCT

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : G06F 15/16	A1	(11) International Publication Number: **WO 96/32681** (43) International Publication Date: 17 October 1996 (17.10.96)

(21) International Application Number: PCT/US95/15483

(22) International Filing Date: 29 November 1995 (29.11.95)

(30) Priority Data:
08/421,566   13 April 1995 (13.04.95)   US

(71) Applicant: CRAY RESEARCH, INC. [US/US]; 655A Lone Oak Drive, Eagan, MN 55121 (US).

(72) Inventors: THORSON, Gregory, M.; 1119 Sweet Water Close, Altoona, WI 54720 (US). SCOTT, Steven, L.; 211 McKinley Avenue, Eau Claire, WI 54701 (US).

(74) Agent: VIKSNINS, Ann, S.; Schwegman, Lundberg & Woessner, 3500 IDS Center, 80 South Eighth Street, P.O. Box 2938, Minneapolis, MN 55402 (US).

(81) Designated States: CA, JP, European patent (AT, BE, CH, DE, DK, ES, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Published
With international search report.

---

(54) Title: ADAPTIVE ROUTING MECHANISM FOR TORUS INTERCONNECTION NETWORK

(57) Abstract

A routing mechanism includes two acyclic non-adaptive virtual channels having two types of virtual channel buffers to store packets along deterministic virtual paths between nodes in an n-dimensional networked system, and an adaptive virtual channel having a third type of virtual channel buffer to store the packets along non-deterministic virtual paths between the nodes. The packets are routed between the nodes along either selected portions of the deterministic virtual paths or selected portions of the non-deterministic virtual paths based on routing information such that a packet is never routed on a selected portion of one of the non-deterministic virtual paths unless the third type virtual channel buffer associated with the selected portion of the one non-deterministic virtual path has sufficient space available to store the entire packet.

30

38
40
42

PROCESSING ELEMENT

ADAPTIVE ROUTING MECHANISM
FOR TORUS INTERCONNECTION NETWORK

5

Field of the Invention

The present invention relates generally to the field of high-speed digital data processing systems, and more particularly, to a method of adaptive routing of data within a multiprocessing network which avoids deadlock.

10

Background of the Invention

Multiprocessor computer systems comprise a number of processing element nodes connected together by an interconnect network. The overall performance of the multiprocessor system depends to a large extent upon the
15 interconnect network performance.

The primary cause of interconnect network performance degradation is congestion due to non-uniform communication patterns and random equations in the network. Even when congestion is limited to a very localized set of processing element nodes, the congestion can lead to a significant backup of
20 messages. One technique that has been used to deal with the congestion problems is adaptive routing. Adaptive routing allows packets of data to be dynamically routed around congested areas in the network. Nevertheless, the potential performance advantages of adaptive routing mechanisms add design complexity and introduce the possibility of deadlock if not implemented
25 properly. Furthermore, in a 3-D torus interconnect topology, previous solutions to avoid deadlock have been excessively expensive.

The interconnect network transmits packets of information between nodes. Examples of information possibly contained in a packet are messages. a shared-memory operation, or various forms of data. Packets comprise
30 multiple physical transfer units (phits). A phit is typically the width of a network physical communication link or physical channel between processing element nodes.

2

Network performance is a function of packet latency and throughput. Packet latency is the time from initiating a packet send until the last phit of the packet is received at its target. The packet throughput is typically measured as the bytes per second per processing element delivered by the

5   interconnect network. Latency and throughput are interdependent and are both dependent upon the communication workload.

Types of Interconnect networks

First generation multiprocessor systems utilize store-and-forward

10   interconnect networks. Store-and-forward interconnect networks transfer packets as single units from node to node along the path from source to destination. Each node waits to pass the head of a packet onto the next node until the last phit of the packet has been received.

More recent multiprocessor systems utilize interconnect networks using

15   some form of wormhole routing. Wormhole routing interconnect networks route the head of the packet from a node before the tail of the packet is received by that node. The packet is divided into a number of smaller message packets called flow control units (flits), which may be one or more phits. A header flit contains routing information. The header flit is received

20   by a processing element node and examined as to its destination. The header flit is sent on to the next node indicated by the routing algorithm. The remaining flits follow behind the header flit in a train-like fashion. Flow control between nodes is accomplished on a flit-by-flit basis, rather than a packet-by-packet basis as in the store-and-forward interconnect networks.

25   Thus, in wormhole routing, a packet may be partially transmitted across a physical communication link, and then blocked due to a shortage of buffer space in the receiving node.

Worm hole routing significantly reduces packet latency in lightly loaded networks, because the time to transmit the packet onto a link (phits per

30   packet times clock period) is suffered only once per network transversal, rather than once per hop. Wormhole routing also significantly reduces

network buffering requirements, as a node is not required to buffer an entire packet.

A problem with wormhole routing, however, is that when a header flit blocks, the remaining flits stall behind the header. These remaining flits may

5   possibly be across multiple links and nodes in the network. A blocked packet may prevent other packets from proceeding, even those that do not want to route through the node at which the header flit is blocked. This can cause significant network degradation, especially in the presence of non-uniform communication patterns.

10   A third type of interconnect network is a virtual cut-through network. A virtual cut-through interconnect network is similar to wormhole routing networks, except that when a packet blocks a buffer, that buffer must always accept the entire packet. Thus, virtual cut-through routing avoids the problems inherent in the blocking occurring in wormhole routing, but at the

15   cost of additional hardware necessary to buffer all of the blocked packets.

The present application is related to mechanisms for enhancing wormhole routing networks.

Deadlock

20   Deadlock occurs when cyclic dependencies arise among a set of channel buffers, causing all involved buffers to fill up and block. A primary consideration in the design of interconnect networks and corresponding routing algorithms is avoiding deadlock.

Deadlock situations can be formalized via a channel dependency graph,

25   a directed graph whose nodes represent network channels and whose arcs represent dependencies between channels. An arc exists between channels x and y iff a packet can route directly from channel x to channel y. It can be proven that a network is deadlock free if its channel dependency graph is acyclic.

30   One simple method to avoid deadlock is to restrict the topology of the interconnect network and/or the routing function used to route packets between the processing element nodes on the interconnect network to remove

4

the possibility of cyclic buffer dependencies. For example, a binary hypercube topology is deadlock-free if the routing function is restricted so that the dimensions are always traversed in increasing order using the e-cube or dimension order routing algorithm. Since at most one hop is made per

5   dimension and no packets route to a lower dimension, there can be no cyclic buffer dependencies. The e-cube routing algorithm can also be used to make an n-dimensional mesh topology deadlock-free, since the opposite-flowing traffic in each dimension uses distinct sets of buffers and the dimensions are traversed in increasing order. The torus topology, however, is not deadlock

10   free when restricted to e-cube routing, because the wrap-around links in the torus topology allow cyclic buffer dependencies to form on a single ring.

In addition, even in meshes, deadlock can arise due to dependencies between request and response packets. Since a node may not be able to accept more request packets until that node has transmitted response packets

15   for previous requests, deadlock can occur if response packets are made to wait behind request packets in the network. An expensive solution to this dependency problem between request and response packets is to use separate physical networks for requests and responses.

20   <u>Virtual Channels</u>

Virtual channels have been used to avoid deadlock and to reduce network congestion. Each physical channel is broken up into one or more virtual channels. Each virtual channel includes virtual channel buffers to store packets along a virtual path. The virtual channels are multiplexed across

25   common physical channels, but otherwise operate independently. Thus, a blocked packet on a first virtual channel multiplexed across the common physical channel does not block packets behind a second virtual channel multiplexed on the common physical channel.

30   <u>Adaptive Routing</u>

Adaptive routing has been used to increase multiprocessor computer system performance. Adaptive routing interconnect networks dynamically

5

route packets around congestion in the network. Thus, adaptive routing
mechanisms dramatically increase network throughput and lower the
sensitivity of the network to variations in communication patterns.

     Adaptive routing algorithms are characterized as being either minimal

5   or non-minimal. Minimal routing algorithms allow only shortest-distance
routing paths between a source node and a destination node. Non-minimal
algorithms allow packets to route along alternate paths that increase the total
routing distance between the source and destination nodes. Thus, non-
minimal algorithms permit adaptive routing in situations where minimal

10   algorithms are constrained to a single path. In this way, non-minimal routing
is used to dynamically route around faults in a network. However, non-
minimal routing causes network interference between processes in different
physical partitions. In addition, non-minimal routing permits livelock
situations to occur, because forward progress is not guaranteed. Deadlock

15   avoidance becomes more complicated with non-minimal routing.

    Minimal adaptive routing algorithms can be fully adaptive or partially
adaptive. Fully adaptive routing allows any of the alternate paths between
source and destination, while partially adaptive routing restricts the choices in
some manner. For example, a partially adaptive algorithm might require that

20   routing in the positive direction of all dimensions be completed before any
routing in the negative direction takes place. In a 2D torus, this results in full
adaptability for a packet moving in the +X and +Y directions, but results in
deterministic routing for a packet moving in the +X and -Y directions.

    Because deadlock must be avoided, adaptive routing algorithms are

25   difficult to implement. As mentioned above, deadlock can be avoided by
restricting the topology or routing function in order to remove cycles from the
channel dependency graph. By contrast, the goal of adaptive routing is to
increase the freedom of the routing function in order to provide a greater
number of alternate paths over which a packet may route.

30   An additional problem introduced by adaptive routing relates to
network ordering. With deterministic (non-adaptive) routing, two packets sent
from the same source to the same destination will be delivered in order. In

6

many situations, the packets must be delivered in a set order such as write requests to the same remote memory location. Adaptive routing in general allows the two packets to take different routes and arrive in the opposite order. Thus, the problem of delivering the packets in order must be addressed

5   if order delivery is required.

Fully Adaptive Routing Using Virtual Channels

In C.R. Jesshope, P.R. Miller and J.P. Yantchev, High Performance Communications in Processor Networks, proc. 16th International Symposium

10   on Computer Architecture, at 150-157, May 1989, a physical network is split into multiple virtual networks, depending upon the direction of travel, in order to break cycles caused by turns. Additional virtual channels within each virtual network are disclosed to break cycles caused by wrap-around links in a torus. 2^n virtual networks are required for an n-dimensional torus or mesh.

15   Double the number of virtual channels are required to break the wrap-around cycles in a 2D torus topology. The Jesshope et al. document does not address cycles created by requests and the corresponding responses. The requests/response cycles double the number of virtual channels required to break all cycles. In order to construct a 3D mesh for request and response

20   traffic using the Jesshope et al. scheme, 16 virtual channels per physical channel would be required. At least 32 virtual channels would be required per physical channel for a 3D torus topology.

In D.H. Linder and G.C. Harden, An Adaptive and Fault Tolerant Wormhole Routing Strategy for k-ary n-cubes, I.E.E.E. Trans. on Computers,

25   at 2-12, January 1991, virtual networks based upon the direction of travel are used for n-dimensional torus or mesh. Linder et al. requires $2^n - 1$ virtual networks for an n-dimensional torus or mesh. In addition, n + 1 virtual channels per virtual network are required to break cycles caused by wrap-around links in a torus. Again, requests/response cycles are not addressed in

30   the Linder et al. reference. A 3D torus topology using the Linder et al. scheme would require 32 virtual channels per physical channel.

In W.J. Dally and H. Aoki, Adaptive Routing Using Virtual Channels,
I.E.E.E. Transactions on Parallel and Distributed Systems, Vol. 4, No. 4, at
466-475, April 1993, both a static algorithm and a dynamic routing algorithm
are presented. In the Dally et al. static algorithm, rather than creating classes

5   of virtual channels based upon the directions of travel, classes of virtual
channels are created based upon the number of dimension reversals a packet
has made. A dimension reversal occurs when a packet routes from a higher
numbered dimension to a lower numbered dimension. With n + 1 such
classes, a packet may only make n dimension reversals and then must route

10 deterministically. The Dally et al. reference only discusses meshes. The
Dally et al. dynamic scheme is discussed below under the deadlock-free
subnetwork section.

Partially Adaptive Routing Using Virtual Channels

15   The above references describe fully adaptive routing mechanisms based
on additional virtual channels. In C.J. Glass and L.M. Ni, The Turn Model
for Adaptive Routing, Proc. 19th International Symposium on Computer
Architecture, at 278-287, May 1992, a turn model routing method is described
which provides partial adaptivity without adding virtual channels. The turn

20 model restricts the turns that packets can make in order to break cycles in the
channel dependency graph. Glass et al. presents several partially adaptive
routing algorithms for 2D mesh topologies. For example, a west-first
algorithm is described which routes packets deterministically west, and then
adaptively in the east, north, and south directions. For minimal routing, the

25 west-first routing algorithm provides full adaptivity for packets that do not
travel west and no adaptivity for those that travel west. The Glass et al.
reference states that it is impossible to provide adaptive, deadlock-free,
minimal routing algorithms for tori with radix greater than four without
adding extra virtual channels. Some non-minimal partially adaptive

30 algorithms for tori topologies are presented in the Glass et al. reference.
        In A.A. Chien and J.H. Kim, Planar-Adaptive Routing: Low-Cost
Adaptive Networks for Multiprocessors, proc. 19th International Symposium

8

on Computer Architecture, at 268-277, May 1992, a planar adaptive model is described. The planar adaptive model provides partial adaptability for tori and meshes by using additional virtual channels. The planar adaptive model routes packets adaptively in a series of 2D planes (e.g., XY and YZ in a 3D

5   network). The planar adaptive model requires a constant number of virtual channels for n-dimensional networks, independent of n, for n greater than two. The planar adaptive model requires three virtual channels per physical channel for a mesh topology and six virtual channels per physical channel for a torus topology. The Chien et al. reference, however, neglects to take account of

10   request/response cycles. A 3D torus network using the planar adaptive model would require 12 virtual channels per physical channel to break the request/response cycles.

Deadlock-Free Subnetwork

15   As mentioned above, it had been previously believed that in order to avoid deadlock in an interconnect network, the channel dependency graph must be acyclic. The following two routing schemes are based on the idea that the channel dependency graph does not have to be acyclic, so long as there is an acyclic portion of the graph which packets can always enter and

20   which allows the packets to reach their destination.

             The first such scheme is the Dally et al. dynamic routing algorithm referred to above and described in the Dally et al. reference. Like the Dally et al. static algorithm, the Dally et al. dynamic algorithm counts the number of dimension reversals a packet takes, but does not limit this number other

25   than by the size of the field and the packet that is used to record the number. The network is broken up into only two classes: adaptive and non-adaptive. All packets start in the adaptive virtual network, and may turn into lower dimensions as often as desired. Any packet that would have to wait on a buffer held by a packet with a lower dimension-reversal number, must leave

30   the adaptive virtual network and route the remainder of the way to its destination in the non-adaptive virtual network. This dynamically prevents deadlock cycles from occurring in the adaptive virtual network.

9

The Dally et al. dynamic routing algorithm provides fully adaptive routing with only a single extra virtual channel per physical channel. However, when a packet must leave the adaptive virtual network to route in the non-adaptive virtual network, the packet must remain in the non-adaptive

5   virtual network for the duration of its travel. Therefore, as soon as a packet drops into the non-adaptive virtual network, adaptability is lost. The Dally et al. dynamic scheme is fairly complex to implement, because the dimension-reversal numbers must be maintained for all packets, and nodes must know the dimension-reversal numbers of packets stored in the channel buffers for all

10   adaptive channels on which the nodes can transmit.

The second adaptive routing scheme using the underlying idea that the channel dependency graph does not have to be acyclic is disclosed in J. Duato, A New Theory of Deadlock-Free Adaptive Routing in Wormhole Networks, I.E.E.E. Transactions on Parallel and Distributed Systems, Vol. 4,

15   No. 12, at 1320-1331, December 1993. The Duato adaptive routing scheme splits the set of virtual channels (C) into two subsets (C1, NC1). The first subset (C1) implements non-adaptive, deadlock-free routing among all nodes. The second subset (NC1) implements minimal adaptive routing and may possibly have cycles in its channel dependency graph. Packets can route

20   freely in the NC1 channels, and circular dependencies may arise among the NC1 buffers. Packets in NC1 must always be able to route into a C1 channel in order to break these cyclic dependencies. Packets in C1 may route to an NC1 channel at any time.

In order to prevent deadlock from occurring, Duato imposes several

25   restrictions in his adaptive routing scheme. First, Duato defines the routing as a function of the current node and the destination node, rather than the more flexible function of the current channel and the destination node. Most routing algorithms define the routing function as a function of the current channel and the destination node. Duato's second restriction is that virtual

30   channel queues (buffers) cannot contain flits belonging to different packets. To implement this requirement, Duato requires that after a queue accepts a tail flit, the queue must be emptied before accepting another header flit. This

10

restriction guarantees that all packets in NC1 will be able to route into C1 if needed to break a cyclic dependency. In the Duato scheme, it is not permitted possible for a head of a packet to be blocked behind another packet in its NC1 queue, so the head is always accessible (at the front of the queue).

5   The third restriction of the Duato scheme is that the extended channel dependency graph of the C1 virtual network must be acyclic. The extended channel dependency graph is the regular channel dependency graph augmented with additional arcs for indirect dependencies. An indirect dependency exists between C1 channels x and y, iff a packet can route from channel x to an

10   NC1 channel, route zero or more additional hops in the NC1 channel, and then route to C1 on channel y. The extended channel dependency graph for the C1 subnetwork of a torus is cyclic. Thus, the Duato scheme does not work for torus topologies.

       None of the above approaches to adaptive routing are suitable

15   mechanisms for 3-D torus networks. Thus, a fully adaptive routing mechanism is desired which provides deadlock free routing on a torus of any dimension and requires minimal virtual channels for implementation.

Summary of the Invention

20   The present invention provides a method and apparatus for routing a packet between a source node and a destination node in a networked system having a plurality of nodes and physical communication links interconnecting the nodes in an n-dimensional topology. A packet is formed by attaching a header having routing information to information to be transferred. Each

25   physical communication link is assigned a first type virtual channel buffer, a second type virtual channel buffer, and a third type virtual channel buffer. Two acyclic non-adaptive virtual channels are defined to be capable of utilizing the first and the second type virtual channel buffers to store the packet along a deterministic virtual path from the source node to the

30   destination node based on the routing information. An adaptive virtual channel is defined to be capable of utilizing the third type of virtual channel

11

buffers to store the packet along a plurality of non-deterministic virtual paths from the source node to the destination node based on the routing information. The packet is routed along either a portion of the deterministic virtual path from the source node to an adjacent node or a portion of one of the non-

5   deterministic virtual paths from the source node to an adjacent node based on the routing information such that the portion of the one non-deterministic virtual path is not selected for routing unless the third type virtual channel buffer associated with the portion of the one non-deterministic virtual path has sufficient space available to store the entire packet. The packet continues to

10   route on virtual paths selected based on the routing information in the above manner until the packet reaches the destination node.

The preferred embodiment of the present invention defines a direction order for routing packets. The direction order defines a priority for routing the packets in each of 2n directions along the deterministic virtual path. The

15   one non-deterministic virtual path is preferably pre-selected for routing in the lowest priority direction of the 2n directions that is not blocked or disabled.

In a preferred embodiment of the present invention a path is selected for routing by pre-selecting the one non-deterministic virtual path and by requesting both the portion of the pre-selected non-deterministic virtual path

20   and the portion of the deterministic virtual path for routing of the packet. Access to the physical communication links is granted or denied in response to the requests and other competing requests for the physical communication links. If an adaptive virtual channel and a non-adaptive virtual channel are both requesting access of a physical communication link the non-adaptive

25   virtual channel is granted access to the physical communication link. The portion of a granted virtual path is selected if at least one of the requests is granted. The portion of the one non-deterministic virtual path is selected if both of the requests are granted. The requests are retried if neither of the requests are granted.

30   The present invention preferably routes multiple packets between the plurality of nodes. In one form of the present invention to support ordered routing, order dependent groups of packets are determined which are required

12

to arrive at the destination node in a pre-determined order. Only the deterministic virtual paths are selected for routing packets which are in an order dependent group of packets.

To account for the cycles created between request packets and
5   response packets, the acyclic non-adaptive virtual channels preferably includes two acyclic non-adaptive virtual channels for handling the request information and two separate acyclic non-adaptive virtual channels for handling the response information.

A preferred embodiment of the present invention uses initial free hops
10   and final hops to provide shorter alternate routes. The initial free hops and final hops permit the interconnect network to configure around many multiple failures that would otherwise be uncorrectable.

Another aspect of the present invention provides a routing mechanism for supporting arbitrary-length packets by allowing the portion of the one non-
15   deterministic virtual path to be selected for routing even though the third type virtual channel buffer associated with the portion of the one non-deterministic virtual path does not have sufficient space available to store the entire packet. This routing mechanism, however, does not select the portion of the one non-deterministic virtual path for routing unless the third type virtual channel
20   buffer associated with the portion of the one non-deterministic virtual path is empty. If a packet is routed to the portion of the one non-deterministic virtual path, a direction and a deterministic virtual path the packet is on as the packet enters the portion of the one non-deterministic virtual path are recorded. When the packet reenters the deterministic virtual path, the packet continues
25   to route in the recorded direction on the recorded deterministic virtual path if all transfers in the recorded direction are not completed.

Brief Description of the Drawings

Figure 1 is a diagram of a massively parallel processing (MPP)
30   system.

Figure 2 is a diagram of a processing element node and interconnect network components for the MPP system of Figure 1.

13

Figure 3 is a diagram of processing element components for the MPP system of Figure 1.

Figure 4 is a diagram of a MPP system having a three-dimensional torus network topology.

5   Figure 5 is a diagram illustrating virtual channels utilized for each network physical communication link or channel of a preferred embodiment of MPP system of Figure 1.

Figure 6 is a block diagram of a switching mechanism in a 3-D network using virtual channels.

10   Figure 7 is a subsection of a MPP system network illustrating a partial Z plane.

Figure 8 is a subsection of a MPP system network illustrating the use of both +X and -X physical communication links to avoid broken physical communication links.

15   Figure 9 is a diagram illustrating a time-of-crossing virtual channel assignment for physical cycles.

Description of the Preferred Embodiments

In the following detailed description of the preferred embodiments,
20   reference is made to the accompanying drawings which form a part hereof, and in which is shown by way of illustration specific embodiments in which the invention may be practiced. It is to be understood that other embodiments may be utilized and structural or logical changes may be made without departing from the scope of the present invention. The following detailed
25   description, therefore, is not to be taken in a limiting sense, and the scope of the present invention is defined by the appended claims.

MPP System

Figure 1 illustrates a simplified model of the components of a
30   massively parallel processing (MPP) system generally indicated at 20. MPP system 20 typically comprises hundreds or thousands of processing element nodes, such as processing element node 22. An interconnect network 24

14

provides communication paths among the processing element nodes in MPP
system 20. In the model of system 20 illustrated in Figure 1, interconnect
network 24 forms a three-dimensional matrix of paths which connect the
processing element nodes in the x, y, and z dimensions as indicated by arrows

5   26. Interconnect network 24 also connects the processing element nodes to an
I/O gateway 28. I/O gateway 28 transfers system data and control
information between a host system (not shown) and MPP system 20.

The host system runs software compilers for MPP system 20. All
software programs written for MPP system 20 are compiled on the host

10   system, but are run in MPP system 20. In a single-cabinet configuration, the
host system resides in the same cabinet as MPP system 20. In a multiple-
cabinet configuration, the host system resides in a separate cabinet that is
cabled to MPP system 20.

Figure 2 provides a more detailed illustration of processing element

15   node 22 and the connection of processing element 22 to interconnect network
24. Each processing element node, such as processing element node 22,
typically comprises a processing element 30, a block transfer engine 32, and a
network interface 34. Processing element node 22 may comprise more than
one processing element 30.

20   As illustrated in Figure 3, each processing element 30 comprises a
microprocessor 38, a local memory 40, and support circuitry 42.

MPP system 20 includes a physically distributed memory, because
each processing element contains local memory. Nevertheless, the MPP
system memory is also logically shared. System memory is logically shared

25   because the microprocessor in one processing element can access a memory
of another processing element without involving the microprocessor in that
processing element.

Local memory 40 preferably comprises a dynamic random access
memory (DRAM) to store system data. A low-latency, high-bandwidth data

30   path connects the microprocessor to local memory in processing element 30.

15

Support circuitry 42 extends the control and addressing functions of microprocessor 38. Support circuitry 42 performs data transfers to or from local memory 40.

Referring back to Figure 2, block transfer engine (BLT) 32 is an
5   asynchronous direct memory access controller that redistributes system data. BLT 32 redistributes system data between local memory in processing element 30 and the local memory in remote processing elements. BLT 32 is capable of redistributing data without interrupting processing element 30 or the remote processing elements. In an alternative embodiment of MPP system
10   20, software controls movement of blocks of data between the local memories of the processing elements.

Network interface 34 formats information to be sent over interconnect network 24 to another processing element node or I/O gateway 28. Network interface 34 also receives incoming information from other processing element
15   nodes or I/O gateway 28 and directs the information to processing element 30.

Interconnect network 24 comprises network routers such as network router 36 illustrated in Figure 2. In addition, interconnect network 24 comprises physical communication links in each direction of each dimension. X dimension communication links 44 connect nodes in the x dimension. Y
20   dimension communication links 46 connect nodes in the y dimension. Z dimension communication links 48 connect processing element nodes in the z dimension. Representative communication links in each dimension are illustrated in Figure 1. In addition, Figure 2 illustrates a communication link 44a in the -X direction, a communication link 44b in the +X direction, a
25   communication link 46a in the -Y direction, a communication link 46b in the +Y direction, a communication link 48a in the -Z direction, and a communication link 48b in the +Z direction.

Communication links 44, 46, and 48 transfer data and control information between the network routers in the interconnect network. A
30   communication link typically comprises two unidirectional channels. Each channel in the link preferably contains data, physical unit (phit) type, virtual channel select, and virtual channel acknowledge signals.

16

Data signals preferably carry either request information or response information. Request information contains information that requests a node to perform an activity. For example, a source node may send a request to a designation node to read data from memory in the destination node. This

5   request is sent over one channel in the communication link. Responses contain information that is a result of an activity. For example, after receiving a request for read data, a destination node sends the response back to the source node. The response contains the read data. Requests and responses are preferably logically separated to avoid deadlock. The logical

10   separation of requests and responses is described below.

Interconnect network 24 preferably connects the processing element nodes in a bi-directional torus. Figure 4 illustrates a preferred embodiment of MPP system 20, having a three-dimensional torus network in the x, y, and z dimensions. Each processing element node in the 3-D torus has

15   communication links in both the + and - directions of the x, y, and z dimensions. Several of the other diagrams in the specification show three-dimensional network connections. For clarity, in these other diagrams, the communication links that complete the torus in each dimension are not shown.

In a torus topology, a ring is formed in each dimension where

20   information can transfer from one node, through all of the nodes in the same dimension and back to the original node. Torus networks offer several advantages for network communication, such as increasing the speed of transferring information. Another advantage of the torus network is the ability to avoid bad communication links by sending information the long way

25   around the network.

Information is transferred between the processing element nodes on the communication links in the form of a packet. Each packet preferably comprises a header. The header includes routing information that steers the packet through the network, destination information that indicates which

30   processing element should receive the packet and control information that instructs the processing element that receives the packet to perform an operation. The header may also contain source information that indicates

17

which processing element created the packet and may contain memory address information. A packet optionally comprises a body. The body of a packet typically includes data to be transferred between processing element nodes or system data from or to the host system.

5   The network routers, such as network router 36, at each processing element node transfer packets through the communication links in interconnect network 24 based on the routing information contained in each header of each packet. The header and the body have variable links and transfer over the communication links one phit at a time. MPP system 20 uses wormhole

10   routing as defined in the Background section in which routing information is contained in the header flit, and the remaining flits follow behind it in a train-like fashion. Flow control between nodes is accomplished on a flit-by-flit basis, rather than a packet-by-packet basis, so it is possible for a packet to be partially transmitted across a link and blocked due to a shortage of buffer

15   space in a receiving processing element node. As mentioned above, the preferred embodiment of the present invention is a bi-directional 3-D torus network, which further complicates deadlock situations.

Adaptive Routing Mechanism

20   An adaptive routing mechanism according to the present invention for routing packets through interconnect network 24 utilizes virtual channels multiplexed across common physical communication links, but otherwise operating independently. As with the Duato deadlock-free subnetwork scheme described in the Background section, MPP system 20 utilizes an adaptive

25   routing mechanism using a first subset of virtual channels C1. The C1 virtual channels implement non-adaptive, deadlock-free routing among all nodes. The second type of virtual channel is an NC1 virtual channel. The NC1 channels implement minimal fully adaptive routing. The NC1 virtual channels may have cycles in its channel dependency graph, since packets route freely

30   in the NC1 channels. As in the Duato scheme, if at any time a decision would be made that would deadlock in the NC1 virtual channel, then the

18

packet must be able to jump to the C1 virtual channel. Thus, packets in NC1 must always be able to route into a C1 channel.

Figure 5 illustrates the virtual channels utilized for each network physical channel or physical communication link. Two virtual channels 50
5   and 52 are used for the C1 virtual request network. Virtual channel 50 is a vc0 type virtual channel, and virtual channel 52 is a vc1 type virtual channel. Similarly, the response C1 virtual network comprises two virtual channels 54 and 56. Virtual channel 54 is a vc0 type virtual channel and virtual channel 56 is a vc1 type virtual channel. The third virtual channel, vc, is the
10   adaptive NC1 virtual channel 58, which provides adaptive routing in both the request and response networks. Although the preferred embodiment illustrated in Figure 5 uses one NC1 channel, a separate NC1 channel may be used for both the request and response virtual networks.

Referring to Figure 6, each node has an input and output for each of
15   the six directions (+X, -X, +Y, -Y, +Z, -Z) as well as the local processing element (PE). The block diagram of Figure 6 illustrates switching in the 3D request or response C1 network using virtual channels. Buffers for the virtual channels reside at the processing node inputs at the receiving end of the physical communication link. The sending node on each physical
20   communication link keeps track of the available buffer space for each virtual channel in the receiving processing element node, using flow control information sent on the opposite-direction link.

At any time, a packet in a C1 virtual channel may route to any NC1 virtual channel at the packets's current node that moves the packet closer to its
25   destination (minimally). A packet in an NC1 virtual channel may also route minimally to any NC1 channel at its current node. A packet in an NC1 channel may route to a C1 channel at any time. Upon re-entering the C1 virtual channel network, the packet routes on a physical and virtual channel according to the deterministic C1 routing algorithm, using the current node's
30   virtual channel look-up table. Within either the request or the response C1 virtual channel networks, choosing to begin a route of a packet on the vc0 virtual channel or the vc1 virtual channel and deciding when to change virtual

19

channels during a route of a packet affects both correctness and performance. The primary consideration is correctness (i.e. the allocation of virtual channels between vc0 and vc1 must prevent deadlock).

The following three steps are performed to prevent deadlock in the C1

5   virtual channels. First, request and response packets are routed on separate sets of virtual channels. This breaks any cycles between request and response traffic. Secondly, direction-order routing is preferably used to break any cycles involving multiple directions. In other words, the packets are routed through a predetermined direction priority, such as (+X, +Y, +Z, -X, -Y, -Z).

10 For example, using the direction order specified above, a route for a +X, -Y, +Z destination would route first +X, then +Z, then -Y. Direction order routing is described in detail in co-pending patent application Serial No. 07/983,979, filed November 30, 1992, and entitled "DIRECTION ORDER ROUTING IN MULTIPROCESSING SYSTEM" and assigned to Cray

15 Research, Inc., the assignee of the present invention.

The third step to prevent deadlock is the use two virtual channels (vc0 and vc1) to break cycles within each ring. This third step can be achieved in a variety of ways to optimize performance. Preferred methods of virtual channel allocation are described in detail in the co-pending and commonly

20 assigned U.S. patent application entitled "OPTIMIZED VIRTUAL CHANNEL ALLOCATION IN PARALLEL PROCESSING SYSTEMS," filed on even date herewith to Steven L. Scott and Gregory M. Thorson, the inventors of the present application. The vc0 or vc1 virtual channel is selected when a packet begins routing on a ring via routing information stored in a virtual channel

25 look-up table 37 (shown in Figure 2) based upon the packet's destination on that ring (the coordinate of the destination node in the corresponding dimension). Each processing element node has a small, system-configurable virtual channel look-up table 37 for each of its six directions. Routing in the C1 network is strictly deterministic.

30 The routing mechanism between C1 and NC1 removes many of the restrictions from the Duato scheme described in the Background section. Unlike the Duato scheme, the routing function takes the virtual channel into

20

account. In addition, the routing mechanism of the present invention permits
flits from multiple packets to reside in the same virtual channel buffer.
Finally, the routing mechanism of the present invention allows cycles in an
extended channel dependency graph of the C1 virtual network. Nevertheless,
5   the following restriction is required for NC1 channels:

> No packet may enter an NC1 channel unless the buffer for that
> channel has room to hold the entire packet.

The above restriction of the adaptive routing mechanism of the present
invention achieves two results. First, the restriction prevents deadlock from
10   occurring within the NC1 channels because no packet can block in an NC1
channel with its tail blocking the head of a different channel buffer. Since a
packet whose head is at the front of an NC1 channel buffer can always route
into a C1 virtual channel, deadlock of the NC1 virtual network is impossible.
By contrast, Duato's scheme requires that no packets share the same channel
15   buffer in order to achieve the same result. Duato's restriction prevents
pipelining the transmission of packets across a link, because a second packet
must wait until the packet ahead of the second packet is drained from its
channel buffer before the second packet can begin transmission.

The second result of the above restriction of the present invention is
20   that indirect dependencies between C1 channels no longer exist. Any packet
that enters NC1 can be fully accepted by the first NC1 channel buffer because
no packet may enter an NC1 virtual channel unless the buffer for that channel
has room to hold the entire packet. Therefore, a packet cannot route from C1
into NC1 and back, and then block with the packet's tail consuming buffer
25   space in the first C1 channel buffer. As a result, the extended channel
dependency graph defined by the Duato reference is not relevant to the
present invention's adaptive routing mechanism. Instead, the routing
mechanism of the present invention can be used to route packets through any
interconnect network topology for which a deadlock-free deterministic routing
30   algorithm on the C1 channels can be constructed. Preferably. as described
above, the deadlock-free deterministic routing algorithm is the direction order

21

routing scheme having two virtual channels assigned to the request network
and two virtual channels assigned to the response network.

The routing mechanism used to select which virtual path to use
between two nodes from the source node to the destination node is as follows:

5   A.   If a head of an NC1 channel and a head of a C1 channel
              are both requesting the physical communication link, the C1
              channel is granted access to the physical communication link.

         B.   A given packet has a pre-selected C1 virtual channel
              path between two nodes and one NC1 virtual channel path
10   between two nodes and requests both the pre-selected C1 path
              and the one NC1 path.

         C.   The one NC1 path is selected as the highest order
              direction that is not blocked or disabled.

         D.   The portion of a granted virtual path is selected if at
15   least one of the requests for the pre-selected C1 path and the
              one NC1 path is granted. If both the pre-selected C1 request
              and the one NC1 request are granted, the one NC1 path is
              selected.

         E.   If neither of the requests for the pre-selected C1 path
20   and the one NC1 path are granted, the packet retries both
              requests on the next clock cycle.

In step C above, the highest order direction is selected so that the
adaptive NC1 channel routes directions that would be routed later in the non-
adaptive C1 channel. For example, for a direction order NC1 channel having
25   a priority of +X, +Y, +Z, -X, -Y, -Z, the lowest order direction is +X and the
highest order direction is -Z.

In step D above, if both the pre-selected C1 request and the one NC1
request are granted, the one adaptive NC1 virtual channel path is selected for
two reasons. First, if a packet was granted the NC1 channel, then no other
30   C1 channels were requesting the physical channel requested by the NC1
channel, because of step A. Therefore, it is less likely that some other packet
is in the given packet's way. The second reason is that the highest order

22

directions tend to be less congested. The highest order directions tend to be less congested because of the direction ordering in the C1 channels. Congestion results because of a string of dependencies from the highest order to the lowest order. As dictated by the direction order scheme, routes tend to

5   enter the C1 channel in the lowest order directions and tend to leave the network on the C1 channel in the highest order directions.

Support for Arbitrary-Length Packets

The preferred MPP system 20 has a maximum packet length, such as

10   slightly longer than a cache line, so that the restriction that an NC1 channel buffer have room for the entire packet before a packet can route on that channel is feasible. Nevertheless, another embodiment of the present invention supports adaptive routing with arbitrary-length packets by removing the restriction that an NC1 channel buffer have room for the entire packet

15   before a packet can route on that channel. This embodiment of an adaptive routing mechanism according to the present invention is also useful if channel buffering is very limited, or in any other case where the requirement that a NC1 channel buffer hold an entire packet is not feasible.

In this alternative embodiment, the routing mechanism uses either of

20   two methods to prevent deadlock among the NC1 channels. This first method is to use the Duato scheme restriction that no packets can simultaneously reside in an NC1 channel buffer. The second method is to use hardware to keep track of the head of the last packet to enter each NC1 buffer, and enable that packet to drain from the NC1 buffer to a C1 channel, bypassing the

25   packet in front of it. Either of these solutions allow cyclic dependencies among NC1 channels to be broken by routing packets from an NC1 channel buffer into a C1 channel.

In this embodiment, NC1 buffers are no longer guaranteed to hold an entire packet. Thus, indirect dependencies can arise between two C1 channels

30   via some number of intermediate NC1 channels. This embodiment of the adaptive routing mechanism according to the present invention prevents these indirect dependencies from causing deadlock in the C1 virtual network in a 3-

23

D torus topology by maintaining extra state information in the packets routing in the NC1 channel and by restricting the choice of which virtual channel to use when reentering the C1 channel from the NC1 channel.

Since the C1 virtual network preferably uses direction-order routing,
5   the only way for indirect dependencies to cause deadlock is for a packet to leave C1 traveling in direction i, route in NC1, and then reenter NC1 and continue routing in direction i. This problem is solved by assuring that a packet that reenters C1 traveling in the same direction as it was traveling before it left C1 continues on the same virtual channel and by aligning
10   datelines of all rings in a given direction.

When a packet traveling in direction d on C1 virtual channel c enters the NC1 network, the virtual channel c is saved in the packet header. A bit in the header (called the continuation bit) is set if the packet has additional routing to do in direction d. As the packet routes in the NC1 channel, the
15   continuation bit is cleared if direction d is satisfied. Given the preferred direction order routing algorithm used in the C1 channel, direction d is always the lowest-order unsatisfied direction, so direction d does not have to be explicitly saved in the packet. If and when the packet reenters the C1 channel, if the continuation bit is set, then the packet is continuing to route in
20   direction d, and the saved C1 virtual channel c is used. Otherwise, the packet is starting in a new, higher-order direction, and the virtual channel is selected via table look-up.

This alternative embodiment routing mechanism guarantees that only a single C1 virtual channel is used for all hops by a packet in a given direction.
25   Since the datelines in virtual channel tables are identical for all rings of a given direction, no cyclic dependencies can arise among the C1 virtual channels.

24

Support for Ordered Routing

The order that packets are delivered to a destination node in an interconnect network having an adaptive routing mechanism is not guaranteed. Since packets may traverse any of the alternate paths between nodes in

5   adaptive routing, a second packet may possibly arrive at a destination node ahead of a packet that was transmitted from the same source node before the second packet.

A response packet can always be routed adaptively, as the order of response packet delivery does not affect program correctness. In addition,

10 MPP system 20 preferably supports a relaxed memory consistency model to allow most requests to be delivered in arbitrary order. In some cases, however, this is not acceptable. For example, two writes by the same process to the same remote memory location must be delivered in order if program correctness is to be preserved.

15   The ability to transmit ordered requests on a packet-by-packet basis is supported by MPP system 20. A special ordered bit is set in an ordered request packet to prevent adaptive routing of the packet. In this way, ordered request packets always traverse the same set of virtual channels between any two given nodes. Thus, ordered request packets cannot pass each other in the

20 interconnect network.

Even if the special ordered bit is set in all request packets indicating that all requests are ordered, adaptive routing can take place with response packets. Moreover, as a general rule, memory request workloads include more read than write requests, so response traffic tends to be heavier than the

25 request traffic in most cases.

Routing of Packets

MPP system 20 preferably uses absolute addressing in all three dimensions, from source to destination node. Alternatively, MPP system 20

30 uses differential addressing wherein the destination address is calculated at the source node and is decremented/incremented to 0 when it reaches the destination node.

25

As described above, the C1 virtual channel preferably uses direction order routing having a predetermined direction order priority such as (+X, +Y, +Z, -X, -Y, -Z). The preferred embodiment of MPP system 20 permits an exception to this predetermined direction order priority by allowing the initial

5   hop of route to be a free hop in any one of the first three directions of the six directions of the ordering, as long as the free hop is not to a dateline. After the particular initial free hop, the packet is allowed to route normally in remaining directions of the six directions as dictated by the direction ordering on the C1 virtual channel or the adaptive ordering of the NC1 virtual channel

10   as prescribed by the adaptive routing mechanism of the present invention. This "initial" hop is free in the sense that the source of a given packet does not need to include any extra routing information in the packet to accomplish the initial hop. The source node router simply throws the packet in the direction of the initial hop, and the routing starts over at the destination of

15   that hop.

In addition to the initial free hop, the preferred embodiment of the routing mechanism is capable of implementing a final hop in the lowest priority direction if a final hop bit is set in the routing information. This final hop permits better access to partial planes. The final hop is not free because

20   one bit (the final hop bit) is needed to indicate whether or not one hop in the final direction is to be taken.

Referring to Figure 7, MPP system 20 preferably is configurable to allow for partial planes, such as partial Z planes. Partial Z planes are planes perpendicular to the Z dimension. Given the direction order priority of (+X,

25   +Y, +Z, -X, -Y, -Z), packet traffic originating on a partial Z plane can take an initial hop in +Z to a plane neighboring the partial Z plane. The packet traffic routes from the neighboring plane with routing tags that are identical to the corresponding routing tags of packet traffic originating from the neighboring plane. Packet traffic to be routed to a destination node on a partial Z plane. is

30   routed to a plane neighboring the partial Z plane in the +Z direction. From the +Z neighboring plane. a final hop in the -Z direction is taken to reach the partial Z plane. The initial and final hops are taken only if a normal route

does not exist from or to a partial Z plane. For example, in Figure 7, a
packet routing from partial Z plane 60 can make an initial hop to complete Z
plane 62. If the destination node is on partial Z plane 60, the packet can
route to complete Z plane 62, and then take a final hop in the -Z direction to
5   reach partial Z plane 60.

There are two primary reasons for having partial Z planes in very large
systems such as MPP system 20. First, MPP system 20 preferably includes
redundant nodes so that MPP system 20 can be reconfigured to replace broken
nodes with the redundant nodes. Secondly, partial Z planes allow the system
10   to be upgraded one module at a time. For example, in one embodiment of
MPP system 20, one module comprises eight processing element nodes in a
liquid cooled system and comprises four processing element nodes in an air
cooled system.

Referring to Figure 8, the routing mechanism according to the present
15   invention uses the initial and final hops to provide shorter alternate routes.
Moreover, the initial and final hops permit the interconnect network to
configure around many multiple failures that would otherwise be
uncorrectable. For example, a packet can be given an initial direction of +X
while the packet's routing bits indicate a -X, +Y, +Z path. Given the
20   direction order priority of (+X, +Y, +Z, -X, -Y, -Z), in this example a packet
makes a single "initial hop" in the +X direction followed by routes in the +Y
direction, then the +Z direction, and finally the -X direction. In Figure 8, the
+Y physical communication link 64 between processing element node 66 and
processing element node 68 is broken. As is illustrated in Figure 8, a packet
25   takes an initial free hop from node 66 to node 70 in the +X direction. From
node 70, the packet then routes in the +Y direction to node 72 and then in the
-X direction to destination node 68 to thereby avoid the faulty communication
link 64.

The initial free hop cannot be made to a dateline node, otherwise a
30   dependency from vc0 to vc0 could be extended across the dateline. Another
restriction to initial hops is that datelines must be aligned for a given

27

direction, otherwise a dependency from vc0 to vc0 could be extended across the dateline.

Virtual Channel Assignments

5   As previously mentioned, the allocation between vc0 and vc1 on either the request or the response virtual channel networks impacts performance as well as correctness. The balancing of the usage of the virtual channels sharing a physical communication link reduces contention in the network and lowers communication latency and increases network bandwidth. The

10   allocation of the virtual channels must be made to break cycles within each ring to assure that deadlock is avoided in the C1 network.

One method of allocating between virtual channels vc0 and vc1, is the time-of-crossing scheme. Referring to Figure 9, the time-of-crossing scheme starts all packets on virtual channel vc0. One node on the ring (dateline node

15   74) transfers any packets that pass through dateline node 74 on the ring onto virtual channel vc1. Since packets never enter and leave dateline node 74 on virtual channel vc0, no cycles are created among the vc0 buffers. Since packets never travel all the way around the ring, no packet ever enters dateline node 74 on virtual channel vc1, and thus no cycles are created among

20   the vc1 buffers. Furthermore, since no packets ever routes from virtual channel vc1 to virtual channel vc0, no cycles are created involving both vc0 and vc1 buffers. Since no packet should ever enter the dateline node from virtual channel vc1, any packet that attempts to cross the dateline on virtual channel vc1 is swallowed by the hardware and an error is flagged.

25   Another method of allocating between virtual channels vc0 and vc1 is the predicted-crossing scheme. The predicted-crossing scheme does not use a physical dateline node that switches packets between virtual channels. Instead, the predicted-crossing scheme uses a logical dateline node or link to avoid deadlock. In an interconnect network using the predicted-crossing

30   scheme, packets are pre-scheduled such that any packet that will cross the logical dateline node or link given the packet's source node and destination node on the ring uses virtual channel vc0, else the packet uses virtual channel

28

vc1. Packets never switch virtual channels on a given ring. The pre-scheduling based on the logical dateline breaks any cycles among the vc1 buffers. In order for there to be no cycles among the vc0 buffers, packets must never route more than halfway around the ring in the predicted-crossing

5   scheme.

Thus, in the time-of-crossing scheme, the dateline node causes messages traveling on one virtual channel to dynamically switch to another virtual channel. By contrast, in the predicted-crossing scheme the messages must be pre-scheduled on virtual channels so that messages will not cross the

10   logical datelines.

Both the time-of-crossing and the predicted-crossing schemes can be improved by carefully selecting the virtual channels on which packets are routed in order to better balance the virtual channel usage. In another virtual channel allocation scheme referred to as the balanced time-of-crossing

15   scheme, some source nodes on a given ring start their packets on virtual channel vc1 rather than virtual channel vc0 to balance virtual channel usage. These packets remain on virtual channel vc1 if they cross the dateline. Thus, in the balanced time-of-crossing scheme care must be taken in selecting the source nodes that start their packets on virtual channel vc1 so that deadlock is

20   not introduced among the vc1 virtual channels of a ring by this optimization.

The balanced predicted-crossing scheme is a superset of the predicted-crossing scheme that takes advantage of the flexibility introduced by packets that cross neither the dateline nor the reciprocal dateline. In this "don't-care" situation, the virtual channel is chosen in order to maximize the balance

25   between vc0 and vc1 traffic.

The balanced predicted-crossing scheme is able to achieve a better balance than the balanced time-of-crossing scheme. However, the balanced time-of-crossing scheme does not pre-schedule messages. As a result, in the balanced time-of-crossing scheme, the virtual channel assignment information

30   is not carried with the packets, but is instead stored in the interconnect network itself. By contrast, the pre-scheduling required with the balanced predicted-crossing scheme is typically carried with the packet. With the

29

balanced time-of-crossing scheme, the virtual channel allocation is not reloaded with broken node information, because the dateline node physically switches the message between virtual channels vc0 and vc1. Thus, in the balanced time-of-crossing scheme, the virtual channel assignments are

5   typically based on an optimized healthy machine. On the other hand, the pre-scheduling required with the logical dateline node or link typically needs to be reloaded and carried with the packet to take account of broken node information.

         With either the balanced time-of-crossing scheme or the balanced

10   predicted-crossing scheme there are various methods to optimize all unrestrained routes. The unrestrained routes are routes of packets that do not cross the dateline in a balanced time-of-crossing scheme and routes of packets that cross neither the dateline nor the reciprocal dateline in the balanced predicted-crossing scheme. Preferred methods of optimizing all unrestrained

15   routes are described in detail in the above cited, co-pending and commonly assigned U.S. patent application entitled "OPTIMIZED VIRTUAL CHANNEL ALLOCATION IN PARALLEL PROCESSING SYSTEMS," filed on even date herewith to Steven L. Scott and Gregory M. Thorson.

20   Conclusion

         The routing mechanism according to the present invention provides fully adaptive, deadlock-free routing on a torus of any dimension. The routing mechanism of the present invention requires only five virtual channels: two C1 channels for request, two C1 channels for responses, and

25   one NC1 channel. Either physical or logical datelines can be used to avoid deadlock in the C1 channels. The routing mechanism can be easily modified to provide ordered delivery on a packet-by-packet basis when necessary. Finally, the routing mechanism of the present invention can provide the same functionality as above by using a modified mechanism if arbitrary-length

30   packets must be supported.

30

What is claimed is:

1.   A routing mechanism for routing packets containing information to be
transferred between nodes in an n-dimensional networked system, the routing
5   mechanism comprising:
        two acyclic non-adaptive virtual channels having two types of virtual
channel buffers to store the packets along deterministic virtual paths between
the nodes;
        an adaptive virtual channel having a third type of virtual channel
10   buffer to store the packets along non-deterministic virtual paths between the
nodes;
        means for routing the packets between the nodes along either selected
portions of the deterministic virtual paths or selected portions of the non-
deterministic virtual paths, wherein a packet is never routed on a selected
15   portion of one of the non-deterministic virtual paths unless the third type
virtual channel buffer associated with the selected portion of the one non-
deterministic virtual path has sufficient space available to store the entire
packet.

20   2.   A routing mechanism for routing packets containing information to be
transferred between nodes in an n-dimensional networked system, the routing
mechanism comprising:
        two acyclic non-adaptive virtual channels having two types of virtual
channel buffers to store the packets along deterministic virtual paths between
25   the nodes;
        an adaptive virtual channel having a third type of virtual channel
buffer to store the packets along non-deterministic virtual paths between the
nodes;
        means for selecting either a portion of a deterministic virtual path
30   between two nodes along said deterministic virtual path or a portion of a non-
deterministic virtual path between two nodes along said non-deterministic
virtual path, wherein the portion of said non-deterministic virtual path is not

31

selected unless the third type virtual channel buffer associated with the portion of said non-deterministic virtual path is empty; and

means for routing the packets between the nodes along the selected portions of said virtual paths including means for recording a direction and a

5   deterministic virtual path which a packet is on when the packet enters the portion of said non-deterministic virtual path, and means for routing the packet in the recorded direction on the recorded deterministic virtual path if all transfers in the recorded direction are not completed when the packet reenters the deterministic virtual path from the portion of said non-

10   deterministic virtual path.

3.   A method of routing a packet between a source node and a destination node in a networked system comprising a plurality of nodes and physical communication links interconnecting the nodes in an n-dimensional topology,

15   the method comprising:

generating a header including routing information;

attaching the header to information to be transferred in order to form a packet;

assigning a first type virtual channel buffer, a second type virtual

20   channel buffer, and a third type virtual channel buffer to each physical communication link;

defining two acyclic non-adaptive virtual channels capable of utilizing the first and the second type virtual channel buffers to store the packet along a deterministic virtual path from the source node to the destination node based

25   on the routing information;

defining an adaptive virtual channel capable of utilizing the third type of virtual channel buffers to store the packet along a plurality of non-deterministic virtual paths from the source node to the destination node based on the routing information;

30   selecting either a portion of the deterministic virtual path from the source node to an adjacent node along the deterministic virtual path or a portion of one of the non-deterministic virtual paths from the source node to

32

an adjacent node along the one non-deterministic virtual path based on the
routing information, wherein the portion of the one non-deterministic virtual
path is not selected unless the third type virtual channel buffer associated with
the portion of the one non-deterministic virtual path has sufficient space

5   available to store the entire packet;

       routing the packet from the source node to the adjacent node along the
portion of the selected virtual path; and

       continuing to select virtual paths at each node from the source to the
destination node and to route the packet on the selected virtual paths until the

10   packet reaches the destination node.

4.   A method of routing a packet between a source node and a destination
node in a networked system comprising a plurality of nodes and physical
communication links interconnecting the nodes in an n-dimensional topology,

15   the method comprising:

       generating a header including routing information;

       attaching the header to information to be transferred in order to form a
packet;

       assigning a first type virtual channel buffer, a second type virtual

20   channel buffer, and a third type virtual channel buffer to each physical
communication link;

       defining two acyclic non-adaptive virtual channels capable of utilizing
the first and the second type virtual channel buffers to store the packet along
a deterministic virtual path from the source node to the destination node based

25   on the routing information;

       defining an adaptive virtual channel capable of utilizing the third type
of virtual channel buffers to store the packet along a plurality of non-
deterministic virtual paths from the source node to the destination node based
on the routing information;

30   selecting either a portion of the deterministic virtual path from the
source node to an adjacent node along the deterministic virtual path or a
portion of one of the non-deterministic virtual paths from the source node to

33

an adjacent node along the one non-deterministic virtual path based on the routing information, wherein the portion of the one non-deterministic virtual path is not selected unless the third type virtual channel buffer associated with the portion of the one non-deterministic virtual path is empty;

5   routing the packet from the source node to the adjacent node along the portion of the selected virtual path;

if a packet is routed to the portion of the one non-deterministic virtual path, recording a direction and a deterministic virtual path the packet is on as the packet enters the portion of the one non-deterministic virtual path, and

10   when the packet reenters the deterministic virtual path continue routing the packet in the recorded direction on the recorded deterministic virtual path if all transfers in the recorded direction are not completed; and

continuing to select virtual paths at each node from the source to the destination node and to route the packet on the selected virtual paths until the

15   packet reaches the destination node.

5.   The method of claim 3 or claim 4 further comprising the step of defining a direction order for routing packets, wherein the direction order defines a priority for routing the packets in each of 2n directions along the

20   deterministic virtual path.

6.   The method of claim 5 wherein the one non-deterministic virtual path is pre-selected for routing in the lowest priority direction of the 2n directions that is not blocked or disabled.

25

7.   The method of claim 3 or claim 4 wherein the step of selecting a path for routing comprises the steps of:

pre-selecting the one non-deterministic virtual path;

requesting both the portion of the pre-selected non-deterministic virtual

30   path and the portion of the deterministic virtual path for routing of the packet;

granting or denying access to the physical communication links in response to said requests and other competing requests for the physical

34

communication links, wherein if an adaptive virtual channel and a non-adaptive virtual channel are both requesting access of a physical communication link the non-adaptive virtual channel is granted access to the physical communication link;

5   selecting the portion of a granted virtual path if at least one of said requests is granted, wherein the portion of the one non-deterministic virtual path is selected if both of said requests are granted; and

retrying the requesting step if neither of said requests are granted.

10   8.   The method of claim 3 or claim 4 wherein the method routes multiple packets between the plurality of nodes, and wherein the method further comprising the steps of:

determining order dependent groups of packets which are required to arrive at the destination node in a pre-determined order; and

15   selecting only the deterministic virtual paths for routing packets which are in an order dependent group of packets.

9.   The method of claim 3 or claim 4 wherein the information to be transferred includes request and response information. and wherein the step of

20   defining acyclic non-adaptive virtual channels includes defining two acyclic non-adaptive virtual channels for handling the request information and two acyclic non-adaptive virtual channels for handling the response information.

10.   The method of claim 3 or claim 4 further comprising sending the

25   packet in a free hop from the source node to an adjacent node prior to the selecting step.

11.   The method of claim 3 or claim 4 wherein the step of transferring the packet further comprises sending the packet on a final hop from a node

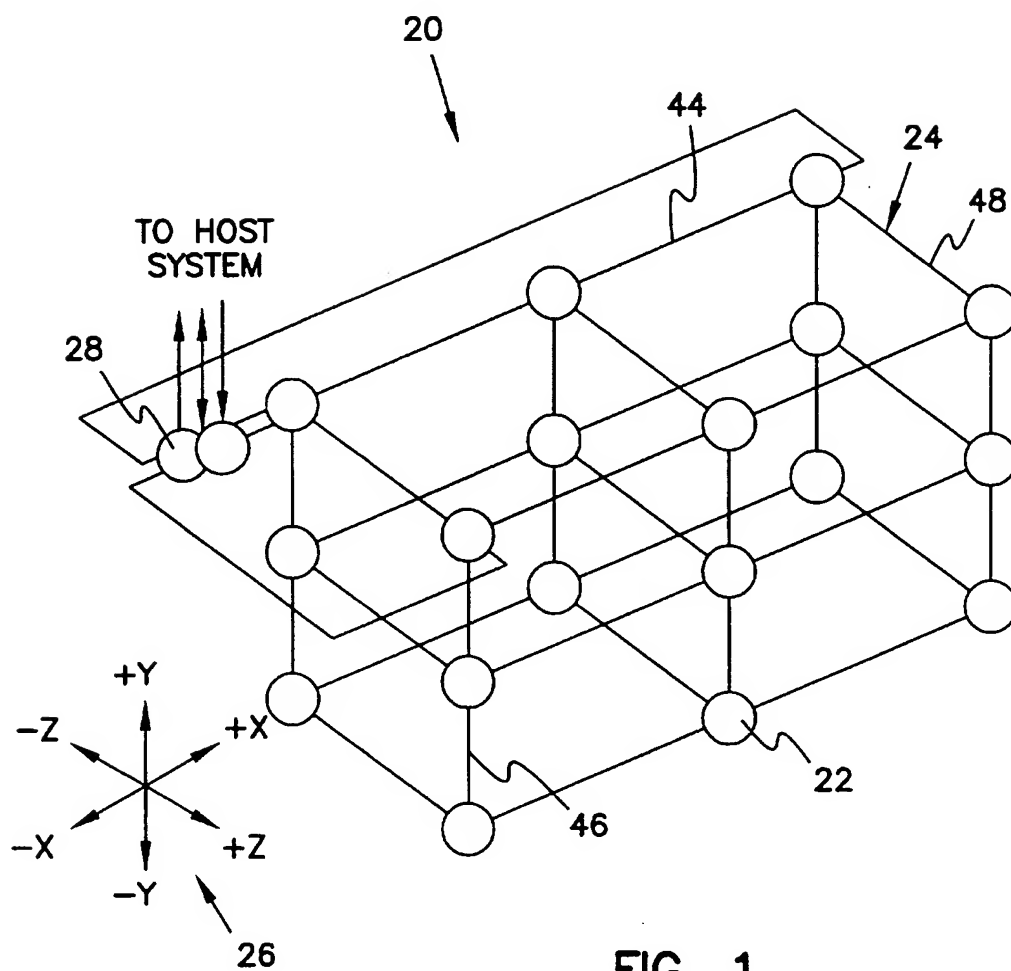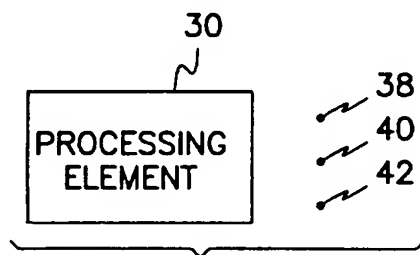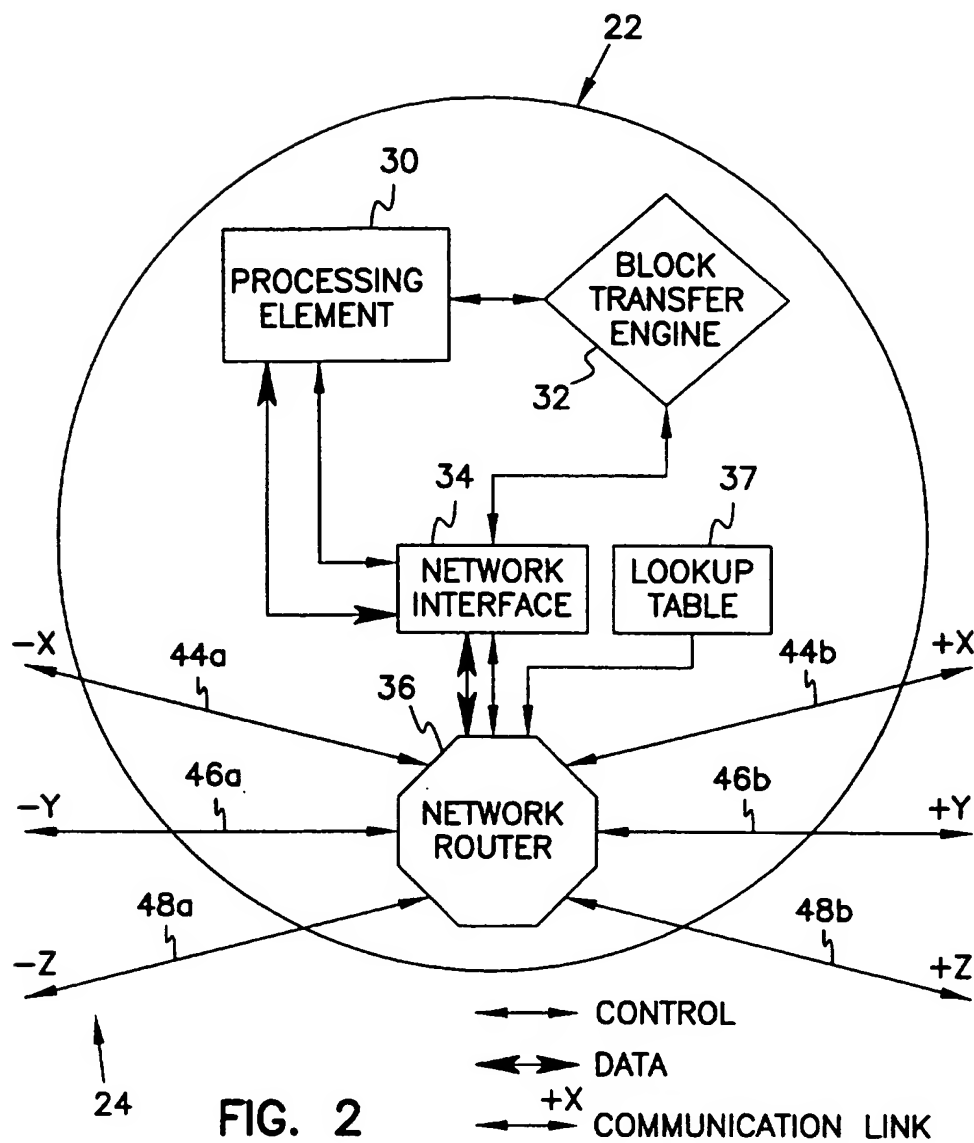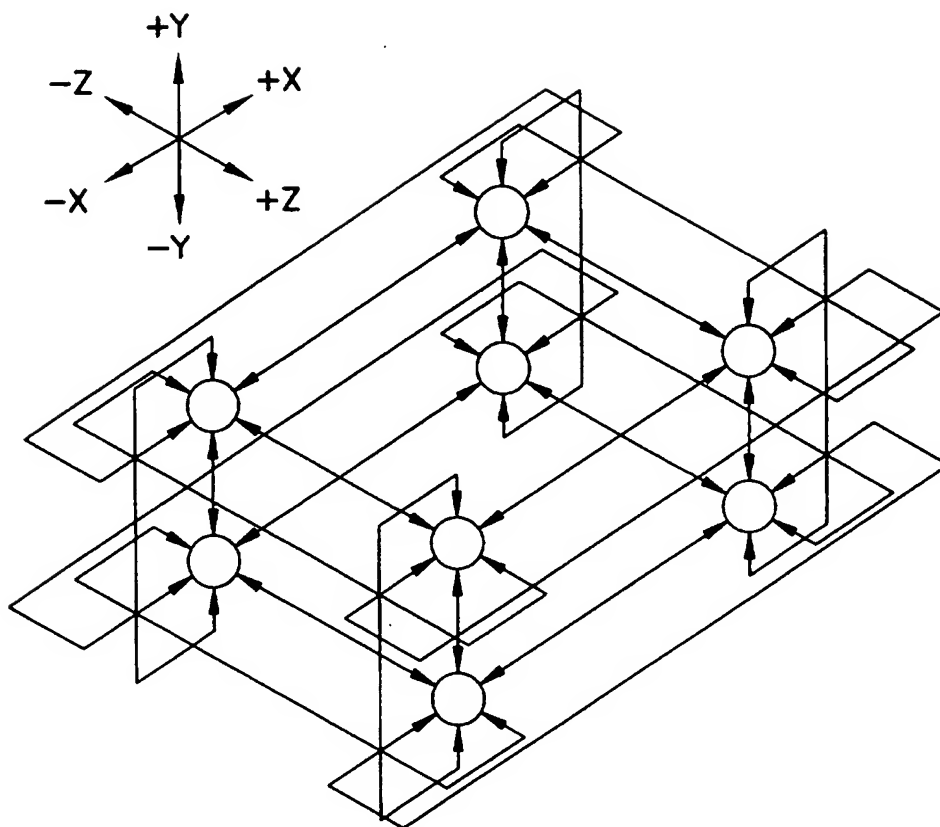30   adjacent to the destination node to complete the route to the destination node.

FIG. 1

FIG. 2

CONTROL
DATA
+X COMMUNICATION LINK

FIG. 3

FIG. 4

REQ C1 vc0 — 50
REQ C1 vc1 — 52
NC1 vc — 58
RESP C1 vc0 — 54
RESP C1 vc1 — 56

PHYSICAL
CHANNEL
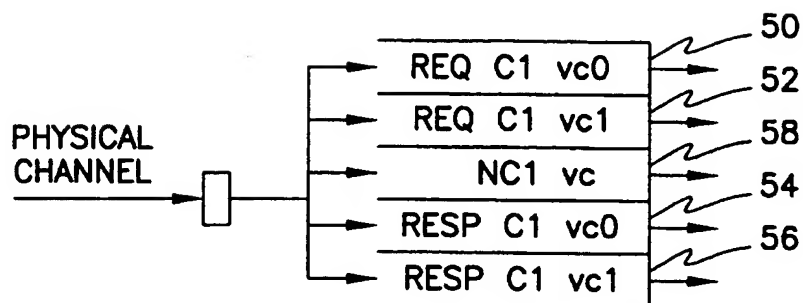
FIG. 5

+X vc0 vc1

+Y vc0 vc1

+Z vc0 vc1

−X vc0 vc1

−Y vc0 vc1

−Z vc0 vc1

PE vc0 vc1

MUX
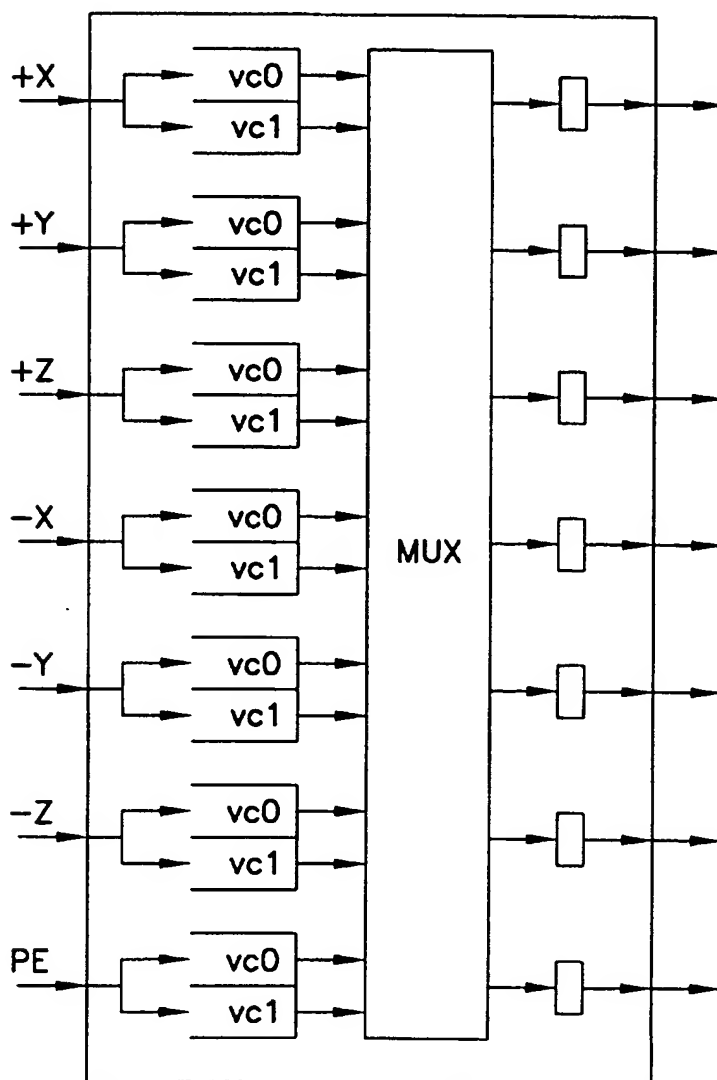
FIG. 6
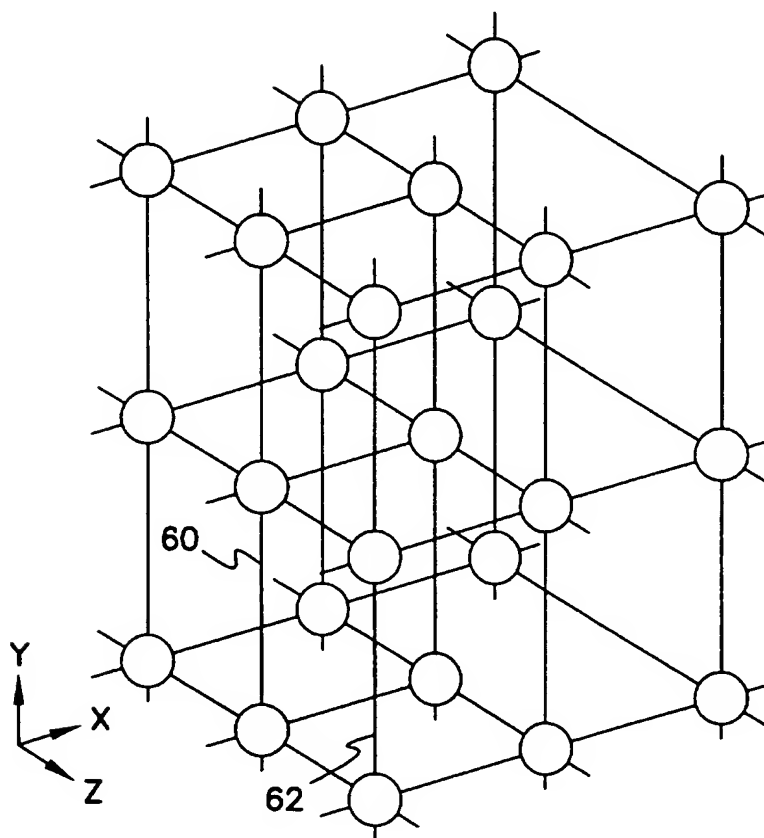
FIG. 7
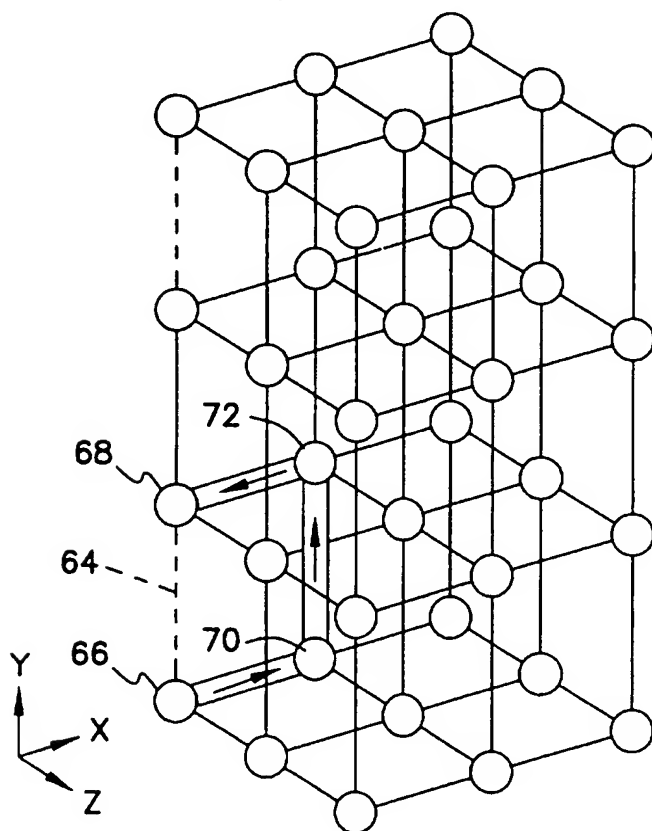
FIG. 8

FIG. 9

INTERNATIONAL SEARCH REPORT

A. CLASSIFICATION OF SUBJECT MATTER
IPC 6   G06F15/16

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 6   G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US,A,4 933 933 (DALLY WILLIAM J ET AL) 12 June 1990 see the whole document ---	1-4
A	TRANSPUTER RESEARCH AND APPLICATIONS 2. NATUG-2 PROCEEDINGS OF THE NORTH AMERICAN TRANSPUTER USERS GROUP, DURHAM, NC, USA, 18-19 OCT. 1989, 1990, AMSTERDAM, NETHERLANDS, IOS, NETHERLANDS, pages 139-177, XP002001410 SHUMWAY M: "Deadlock-free packet networks" see abstract see page 156, line 10 - line 27 --- -/--	1-4

[X] Further documents are listed in the continuation of box C.   [X] Patent family members are listed in annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
24 April 1996	0 9. 05. 96

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (- 31-70) 340-2040, Tx. 31 651 epo nl, Fax (- 31-70) 340-3016	Schenkels, P

Form PCT-ISA/210 (second sheet) (July 1992)

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT		
Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	PARLE '93 PARALLEL ARCHITECTURES AND LANGUAGES EUROPE. 5TH INTERNATIONAL PARLE CONFERENCE PROCEEDINGS, MUNICH, GERMANY, 14-17 JUNE 1993, ISBN 3-540-56891-3, 1993, BERLIN, GERMANY, SPRINGER-VERLAG, GERMANY, pages 254-265, XP002001411 LIU Z ET AL: "Grouping virtual channels for deadlock-free adaptive wormhole routing" see abstract	1-4

A	PROCEEDINGS OF THE 14TH INTERNATIONAL CONFERENCE ON DISTRIBUTED COMPUTING SYSTEMS , POZMAN, POLAND, 21-24 JUNE 1994, ISBN 0-8186-5840-1, 1994, LOS ALAMITOS, CA, USA, IEEE COMPUT. SOC. PRESS, USA, pages 589-596, XP002001412 BOURA Y M ET AL: "Efficient fully adaptive wormhole routing in n-dimensional meshes" see page 589, left-hand column, line 1 - page 594, left-hand column, line 15; figures 1-5	1-4

A	IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, DEC. 1994, USA, vol. 5, no. 12, ISSN 1045-9219, pages 1233-1251, XP002001413 GRAVANO L ET AL: "Adaptive deadlock- and livelock-free routing with all minimal paths in torus networks" see abstract	1-4

1

INTERNATIONAL SEARCH REPORT

information on patent family members

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US-A-4933933	12-06-90	NONE	